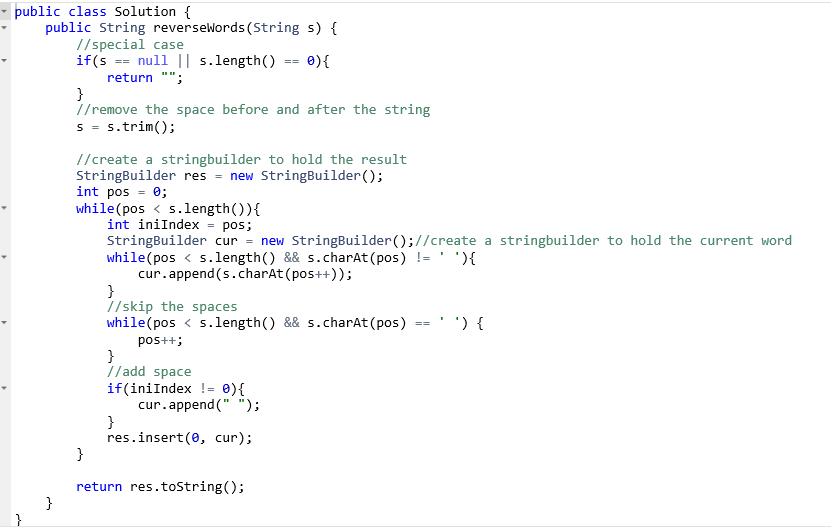
**1 Reverse Words in a String**

For this problem, I think we can use stringbuilder class to construct the result because we can insert strings in the beginning of a stringbuilder. The basic algorithm is that we scan the original string, recognize each word in it and insert them into the stringbuilder from the end to the beginning one by one. However, we should pay attention to some problems like: there are spaces in the beginning or the end, there are extra spaces between words. In the program, first we handle the special case that the input string is null or its length is zero. Then, we use trim() to eliminate all the spaces in the beginning or the end of the string. As I said, we use a stringbuilder to store the results. We need to scan the string and I use “pos” as the index. I use a while loop and the condition is Pos < s.length(). To handle the problem of adding space after each word, we need to store the index of the beginning of each new word. A word is a string sequence that does not contain space, thus I use another while loop to extract each word. If the current character is not space, we append it to a new stringbuilder called cur. After it I use another while loop to eliminate all the spaces after the word. If the beginning of the current index is not zero, we should add a space after the word, then we insert the current word into the beginning of the result. When the outer while loop exits, we convert res to string and return.



时间复杂度：O(n), 空间复杂度O(n)

**2 Reverse Words in a String II**

For this problem, we are required to solve the problem in place. Thus, we must adopt a new algorithm: First, we reverse the whole character array. Then, we reverse each word one by one. After these two reversions, we can get what we want. I create a function called reverse, it takes three inputs, the character array itself and two index: begin and end. This function reverses the specific part of the char array, from begin to end. In this function I use a while loop, when begin is smaller than end, we switch the two characters, then begin plus one and end minus one. In the reverseWord function, we first call reverse to reverse the whole char array. Then, we use a for loop to scan the whole array. There are two indices, I and j. When the character pointed by I is a space, or it reaches the end of the array, we should reverse the word in from of it, which can be indexed by (I to j – 1 or j). Then we add 1 to j, let it point to the beginning of the next word, and let I point to it, then we continue the for loop.

（暂无图片）

时间复杂度：O(n), 空间复杂度O(1)